

WE CLAIM:

1. In a computer system, an improved multi-client to multi-server software system comprising:
 at least one server process software application capable of sending and receiving messages;
 5 at least one client process software application to said server process software application
 capable of sending and receiving messages;
 a control process software module for passing said messages to and from said server process
 and client process.

10 2. The invention of claim 1, wherein:

said server process and said client process send and receive messages only to and from said
 control process software module, and communication between said server process and said client
 process occurs under direction of said control process, said control process acts as a message
 broker between said server process and said client process.

15 3. The invention of claim 2, wherein:

said control process controls the running of said server process and said client process;
 said control process sets synchronization points, said synchronization points comprising
 points in time where said control process pauses the running of said server process.

20 4. The invention of claim 3, further comprising:

a plurality of server processes, a plurality of client processes, and each of said plurality of
 server processes communicating via said control process with a predetermined number of said
 plurality of client processes associated with each of said server processes, with said control
 25 process controlling said plurality of server and client processes.

wherein said control process stops the running of said server process when each of said server
 process reaches a synchronization point, said synchronization points in time being elapsed time
 from the start of simulation by said control process.

30 5. The invention of claim 2, further comprising:

a plurality of client processes associated with said server process, each of said plurality of client processes communicating via said control process with said server process, with said control process controlling said server process and said client processes.

5 6. The invention of claim 2, further comprising:

a plurality of server processes, a plurality of client processes, and each of said plurality of server processes communicating via said control process with a predetermined number of said plurality of client processes associated with each of said server processes, with said control process controlling said plurality of server and client processes.

10 7. The invention of claim 6, wherein:

said control process sets up a predetermined ordered queue of said server processes and a predetermined ordered queue of said client processes, and said messages are sent to and from client and server according to said predetermined ordered queue of server processes and client processes.

15 8. The invention of claim 3, wherein:

said server process evaluates a predetermined event expression to determine the occurrence of an event in said server process, and;

20 at least one said server process sends a event expression message to said control process upon the occurrence of said predetermined event expression in said server process, said event expression message containing a time stamp, said time stamp an indication of the time at which said event occurred in said server process.

25 9. The invention of claim 8, further comprising:

a plurality of server processes, a plurality of client processes, and each of said plurality of server processes communicating via said control process with a predetermined number of said plurality of client processes associated with each of said server processes, with said control process controlling said plurality of server and client processes.

30

10. The invention of claim 9, wherein:

said control process maintains said time stamp for each server, said time stamp being an indication of the time elapsed from the start of the control process, and said time elapsed
5 proportional to the time elapsed in said control process between said synchronization points.

11. The invention of claim 9, wherein:

said control process sets up a server order queue comprising a predetermined ordered queue of said server processes and a client order queue comprising a predetermined ordered queue of
10 said client processes, and said messages are sent to and from client and server according to a predetermined ordered queue comprising said server order queue and said client order queue.

12. The invention of claim 11, wherein:

said control process receives a plurality of said event expression messages from said server
15 processes, and said control process sorts said event expression messages received from said server processes according to the server order queue;

said control process ordering each of said event expression messages within said server order queue according to the earliest time of said time stamp at which said event occurred in said
20 server process.

13. The invention of claim 11, wherein:

said control process delivers said sorted event expression messages to said client processes associated with said server processes according to said predetermined ordered queue of client
25 processes.

14. The invention of claim 5, wherein:

each of said plurality of client processes sends a finish message, indicating said client process is finished running, to said control process for communication to said server process associated with said client process, when each of said client processes is finished running;

said control process holds each of said finish messages from said plurality of client processes until all of said plurality of client processes associated with a server process are finished running; and,

wherein said control process sends a finish message to said server indicating the client processes are finished running.

15. The invention of claim 14, wherein:

each of said plurality of server processes sends a finish message, indicating said server process is finished running, to said control process when said client processes associated with each of said server processes are finished;

said control process holds each of said finish messages from said plurality of server processes until all of said plurality of server processes have sent said finish messages to said control process;

wherein said server processes, client processes, and control process finish operations and exit.

16. The invention of claim 2, further comprising:

a plurality of client processes, said plurality of client processes associated with a predetermined server process, communicating with said server process under the direction of said control process;

a plurality of server processes, each of said server processes evaluates an event expression to determine the occurrence of an event in said server process, and each of said server processes sends an event expression message to said control process upon the occurrence of said event in said server process, said event expression message containing a time stamp indicating the time at which said event occurred in said server process.

17. The invention of claim 16, further comprising:

said control process software module sets up a plurality of predetermined ordered queues comprising a client ordered queue of client applications in a particular order, a server ordered queue of server applications in a particular order, and a time ordered queue of event expression

messages received from said plurality of server applications, said time ordered queue ordered according to the earliest in time event expression message.

18. The invention of claim 16, wherein:

5 said control process software module resides within said server process application, in the code comprising said server process application.

19. A server-client computer simulation system comprising:

a computer comprising a processor, primary and secondary memory, means for I/O;

10 at least one server comprising a processor, primary and secondary memory, means for I/O, and a server application residing in said memory and operating said processor;

at least one client comprising a processor, primary and secondary memory, means for I/O, and a client application residing in said memory and operating said processor;

15 a control process software module residing in said computer memory, said control process software module acting as a message broker between said server application and said client application, for passing messages between said server application and said client application, and communication between said server application and said client application controlled and directed by said control process software module, said server-client computer simulation system acting to simulate a device in a repeatable manner.

20

20. The invention of claim 19, wherein:

said device simulated is a device selected from the group consisting of electrical devices, mechanical devices, electromechanical devices, computer networks, DSL modems, ASICs disk drive controllers, graphics processors, network interface adapters and communications networks.

25

21. The invention of claim 19, wherein:

said control process software module controls said server application and said client application, and said control process sets synchronization points for said server application comprising points in time where said control process software module pauses the running of said
30 server application.

22. The invention of claim 21, wherein:

said control process software module comprises a synchronization varying software module for varying the elapsed time duration between said synchronization points.

5

23. The invention of claim 21, wherein:

said control process stops all of said servers upon said servers reaching a synchronization point.

24. The invention of claim 19, further comprising:

a plurality of client applications, said plurality of client applications associated with said server application, and communicating with said server application under the direction of said control process software module.

25. The invention of claim 24, wherein:

a plurality of server applications, said plurality of server applications communicating via said control process software module with a predetermined number of said plurality of client applications associated with each of said server applications.

26. The invention of claim 25, wherein:

said control process software module sets up a plurality of predetermined ordered queues comprising a client ordered queue of client applications and a server ordered queue of server applications.

27. The invention of claim 21, wherein:

a plurality of server applications, a plurality of client applications associated with said server applications, said plurality of server applications communicating via said control process software module with said predetermined number of said plurality of client applications associated with each of said server applications; wherein,

each of said server applications evaluates an event expression to determine the occurrence of an event in said server application, and each of said server applications sends an event expression

message to said control process software module upon the occurrence of said event in said server application, said event expression message containing a time stamp indicating the time at which said event occurred in said server process.

5 28. The invention of claim 27, wherein:

said control process software module sets of a plurality of predetermined ordered queues comprising a client ordered queue of client applications in a particular order, a server ordered queue of server applications in a particular order, and a time ordered queue of event expression messages received from said plurality of server applications, said time ordered queue ordered according to the earliest in time event expression message, and said control process software module passing messages to and from said server and said client applications according to at least one of said predetermined ordered queues.

10 29. A method of carrying out a simulation employing multiple clients and multiple servers

15 comprising the steps of:

running a plurality of server process software applications that simulate a server application;

running a plurality of client process software applications that each simulate a client application, each of said client applications associated with at least one of said server applications;

20 running a control process software application that acts as a message broker between said servers and clients, all messages between servers and clients managed and controlled by said control process, and said control process controlling the operation of said servers;

maintaining the elapsed time of said simulation in said control process software application.

25 30. The invention of claim 29, further comprising the steps of:

determining the occurrence of a predetermined event in said server applications;

maintaining, in said control process, a list of client applications and server applications, and a list of messages for the occurrence of said predetermined events that occur in said server applications;

communicating said predetermined events from said server applications to said client applications.

31. The invention of claim 30, further comprising the steps of:

- 5 ordering, in said control process, said messages of said predetermined events according to the earliest time that such predetermined events occurred in said server applications; and,
 delivering said messages to said client applications according to said ordering of said predetermined events.

10 32. The invention of claim 30, wherein:

 ordering, in said control process, said list of messages for the occurrence of said predetermined events according to (1) time order, the earliest time that such predetermined events occurred in said server applications, (2) server order, an ordering according to a predetermined queue of servers, and, (3) client order, an ordering according to a predetermined
 15 queue of clients.

33. The invention of claim 32, wherein:

 sorting said list of messages of said predetermined events according to said server order and said time order;

- 20 delivering, using said control process, said messages of said predetermined events from said control process to said plurality of client applications according to said client order and said time order, with the earliest messages delivered first.

34. The invention of claim 29, further comprising the steps of:

- 25 setting a plurality of synchronization points comprising elapsed time in the simulation of servers and clients;
 stopping said servers upon each of said servers reaching said synchronization points.

35. The invention of claim 34 , further comprising the steps of:

varying the duration of elapsed time between said synchronization points by way of said control process setting the duration of time to elapse between synchronization points.

36. The invention of claim 29, further comprising the steps of:

- 5 setting a plurality of synchronization points comprising elapsed time in the simulation of servers and clients;
- determining the occurrence of a predetermined event in said server applications;
- maintaining, in said control process, a list of client applications and server applications, and a list of the occurrence of said predetermined events that occur in said server applications;
- 10 communicating said predetermined events from said server applications to said client applications;
- ordering, in said control process, said predetermined events according to the earliest time that such predetermined events occurred in said server applications; and,
- delivering messages to said client applications relating to said predetermined events
- 15 according to said ordering of said predetermined events.

37. The invention of claim 36, further comprising the steps of:

- 20 determining through said control process whether said client applications are finished with said simulation through the occurrence of a message indicating said client applications are finished;
- determining through said control process whether said server applications are finished with said simulation through the occurrence of a message indicating said server applications are finished;
- 25 said control process acknowledging said client and server application finish messages and said simulation terminating when said client and server applications have all finished.

38. The invention of claim 29, further comprising the steps of:

- polling each of said plurality of client process software applications with said control process software application in a predetermined manner;

temporarily storing said messages from said client process software applications, until such time that said client process software applications issue a predetermined message to simulate to said control process;

forwarding said messages from said client process software applications to said server process software applications associated with said client process software applications upon the occurrence of said predetermined message to simulate. A simulator apparatus comprising:

means for sending and receiving messages in a computer system, said means for sending and receiving messages acting as a server;

means for sending and receiving messages in a computer system, said means for sending and receiving messages acting as a client;

means for sending and receiving messages server means and said client means, said means for sending and receiving messages acting as a message broker between said server means and said client means, and said means for sending and receiving messages able to stop the running of said server means and said client means at predetermined points in time comprising synchronization points; wherein,

said server means, said client means and said message broker means act as a simulator performing a repeatable simulation.

40. The apparatus according to claim 39, wherein:

said server means evaluates a predetermined event expression to determine the occurrence of an event in said server means, and;

said server means sends a event expression message to said message broker means upon the occurrence of said predetermined event expression in said server means, said event expression message containing a time stamp, said time stamp an indication of the time at which said event occurred in said server means;

a plurality of said server means and said client means, wherein said message broker means delivers said event expression messages between said server means and said client means according to a predetermined queue.